

METHOD AND SYSTEM FOR USER-CONFIGURABLE CLUSTERING OF INFORMATION

FIELD OF THE INVENTION

5 This invention relates to the field of pattern processing and information management and more specifically to methods and systems for incorporating user interests and preferences in automated information clustering. Related fields of invention include information database content management, self-organizing information databases, document clustering, and personalized systems.

BACKGROUND OF THE INVENTION

10 Categorization and clustering have been two fundamental approaches to information organization and information database content management.

15 Categorization or classification is supervised in nature. A user defines a fixed number of classes or categories. The task is to assign a pattern or object to one or more of the classes. Categorization provides good control in the sense that it organizes the information according to the structure defined by the user. However, due to the predefined structure, categorization is not well suited to handling novel data. In addition, much effort is needed to build a categorization system. It is necessary to specify
20 classification knowledge in terms of classification rules or keywords (disclosed in U.S. Pat. No. 5,371,807) or to construct a categorization system through some supervised learning algorithms (disclosed in U.S. Pat. Nos. 5,671,333 and 5,675,710). The former requires knowledge specification (*e.g.*, written classification rules) and the latter requires example annotations (*i.e.* labeling information). Both are labor intensive.

25 Clustering is unsupervised in nature. For unsupervised systems (U.S. Pat. Nos. 5,857,179 and 5,787,420), there is no need to train or construct a classifier since information is organized automatically into groups based on similarities. However, a user has very little control over how the information is grouped together. Although it is possible to fine tune the parameters of the similarity measures to control the degree of
30 coarseness, the effect of changing a parameter cannot be predicted; changing one

parameter could affect all clustered results. In addition, the structure established through the clustering process is unpredictable. Whereas clustering is acceptable for a pool of relatively static information, in situations where new information is received every day, information with similar content may be grouped (based on different themes) into different clusters on different days. This ever-changing cluster structure is highly undesirable for the user who is navigating the information database to find desired information. Imagine the frustration of reading a newspaper with a different layout every day! U.S. Pat. No. 5,911,140 attempts to provide a solution by ordering document clusters based on user interests. However, the cluster ranking relies on the availability of the ranking of each document in the clusters and only very minimal user preferences are taken into account.

SUMMARY OF THE INVENTION

In order to overcome the various shortcomings of systems which effect only categorization or clustering of information with little or no account taken of user preferences, this invention provides a method and system that incorporate users' preferences in an information clustering system. In general, this system allows a user to create a cluster structure and influence or personalize the cluster structure by indicating his or her own preferences as to how information should be grouped. This invention further allows the user to store the cluster structure and subsequently retrieve it for future use.

The user-configurable information clustering system comprises an *information clustering engine* for clustering of information based on similarities, a *user interface module* for displaying the information groupings and obtaining user preferences, a *personalization module* for defining, labeling, modifying, storing and retrieving cluster structure, and a *knowledge base* where a user-defined cluster structure is stored.

According to the invention, each unit of information is represented by an *information vector*. A user preference, indicating a preferred grouping for the corresponding unit of information, can be represented by a *preference vector*. In addition, information, which may be in the form of a database, is supplied to the user-configurable information clustering system by any well known means within the art.

In the preferred embodiment, the information clustering engine is a hybrid neural network comprising two input fields F_1^a and F_1^b with an F_2 cluster field. The F_1^a field serves as the input field for the information vector **A**. The F_1^b field serves as the input field for the preference vector **B**. The F_2 field contains a plurality of cluster nodes, each encoding a template information vector w_j^a and a template preference vector w_j^b . Given an information vector **A** with an associated preference vector **B**, the system first searches for an F_2 cluster J encoding a template information vector w_j^a that is closest to the information vector **A** according to a similarity function. It then checks if the associated F_2 template preference vector w_j^b of the selected category matches with the input preference vector. If so, the templates of the F_2 cluster J are modified to encode the input information and preference vectors. Otherwise, the cluster is reset and the system repeats the process until a match is found.

Through the user interface and the personalization module, the user is able to influence the cluster structure by indicating his own preferences in the form of preference vectors. The user can create a new cluster, label an existing cluster, and/or modify cluster structure by merging and splitting clusters. In addition, the resulting customized cluster structure can be stored in the *cluster structure knowledge base* and retrieved at a later stage for processing new information.

In one embodiment of the invention, a system and method are provided for customizing the organization of an existing set of information according to the user's knowledge and preferences. In another embodiment, a system and method are provided for creating a cluster structure automatically and subsequently modifying this machine-generated cluster structure according to the user's preferences. In yet another embodiment, a system and method are provided for detecting new information and analyzing trends wherein the user, through repeated personalization of the cluster structure, identifies new information and trends previously unknown to him or her.

The disclosed invention is more flexible than a pure categorization system in which information must be assigned to one or more pre-defined categories or groups. At the same time, it is more flexible than a pure clustering or self-organizing system in which information is grouped according to similarities but the user has very little control over how the information is organized.

The invention has a number of advantages over the prior art: The invention performs clustering or self-organization of information based on similarities in content, *i.e.* similarities in information vectors. The information can be automatically organized without user training or prior construction of a classifier. The invention allows the user to correct or change the organization of the information as necessary. The invention also allows the user to intervene in the organization of the information both globally and locally. Further, the invention allows the user to control the coarseness of the information groupings without tuning the parameters of complex similarity functions. The invention also allows the user to indicate directly how specific units of information are to be organized.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the invention will now be described by way of example with reference to the accompanying drawings in which:

FIG. 1 illustrates an embodiment of a user-configurable information clustering system according to the present invention.

FIG. 2A shows an exemplary architecture for the information clustering engine of FIG. 1 for clustering information in response to user preferences. Information is encoded as information vectors. User preferences are encoded as preference vectors. The F_1^a field serves as the input field for the information vectors. The F_1^b field serves as the input field for the preference vectors. Information clusters (represented as F_2) are formed through the synchronized clustering of the information and preference vectors.

FIG. 2B illustrates the category choice process of the information clustering engine of FIG. 1.

FIG. 2C illustrates the template matching and the template learning process of the information clustering engine of FIG. 1.

FIG. 3 illustrates an exemplary flow diagram for incorporating user preferences in the information clustering process.

FIG. 4 illustrates an exemplary set of personalization functions that a user could use while performing information clustering.

DETAILED DESCRIPTION OF THE INVENTION

The invention is concerned with the organization of information, based on similarities in content, *i.e.* similarities in information vectors, according to a user's knowledge and preferences. The information comprises text, image, audio, video, or any combinations thereof. According to the invention, each unit of information (Φ), defined as an individual element of information, may be any object, for example a document, person, company, country, etc., and can be represented by a complement coded 2M-dimensional information vector A of *attributes* or *features*,

$$A = (a, a^c) = (a_1, \dots, a_M, a_1^c, \dots, a_M^c) \quad (1)$$

where a_i is a real-valued number between zero and one, indicating the degree of presence of attribute i , and $a_i^c = 1 - a_i$. Complement coding represents both the on-response and the off-response to an input vector and preserves amplitude information upon normalization.

In the case of documents, for example, the features in the information vectors could be token words commonly known as *keywords*. For example, in the exemplary case where the information is a business document, keywords may include "share", "market", "stock", "acquisition", "trading" etc. The feature sets can be predefined manually or generated automatically from the information set.

User preferences are represented by *preference vectors* that indicate the preferred groupings of the information. A preference vector B is a complement coded 2N-dimensional vector defined as

$$B = (b, b^c) = (b_1, \dots, b_N, b_1^c, \dots, b_N^c) \quad (2)$$

where b_k is either zero or one, indicating the presence or absence of a user-defined class label L_k , and $b_k^c = 1 - b_k$. A user's knowledge comprises that information acquired by one in the user's profession, community, or field of endeavor or study. Also, this knowledge may include highly specialized information developed, acquired or used based on a user's unique experiences. An example of user knowledge includes a physician's knowledge of diseases, symptoms for those diseases, and appropriate treatments, including appropriate drugs; knowledge of this type from many practitioners could be placed in a database which could increase in content as new diseases were discovered, treated and cured.

Using the invention, a physician could configure the database and additions thereto over time according to her own experience and specialty. Another example of user knowledge is a market analyst's understanding of trends in a given manufacturing sector over a given period of time; a database of wire news reports over a given period or updated on a periodic basis may contain information of use to the market analyst. Using the invention, the market analyst could configure such a database and updates thereto according to the particular sector of interest to him. Yet another example of user knowledge is a chef's knowledge of the cooking arts; a database of culinary dishes, ingredients, foods, etc., which may be intermittently modified, contains information of use to the culinary field.

Using the invention, the chef can, for example, create groupings of ingredients for culinary dishes of interest to her. Another example of user knowledge is the preferred grouping of news articles by a journalist. A journalist, depending on his target readers, may organize articles from local and foreign sources into specific groupings that would be most appropriate for his purposes. For example, a Singapore journalist might organize foreign news into threads that are of interest to Singaporeans, such as "Michael Fay Event", "Singapore Economics" etc. Other examples will be evident to those skilled in the art. In addition, the user's preferences are derived from, for example, his personal or professional informational desires, organizational goals and objectives, analytical training and skills, interpretational biases, and experiences with identification of information. User preferences include informational and organizational objectives useful to organizing and increasing the knowledge base of the user. For example, in the case of the Singapore journalist mentioned above, preferences used in organizing the articles are based on the journalist's interpretation and experience on the relevancy of such topics to his readers and the volume of articles on such topics over time.

Units of information which share common attributes or content, as indicated by their information vectors, are said to be similar to each other. Units of information which share few or no common attributes or content are said to be dissimilar to each other. The system may automatically determine similarity between two units of information based on the application of a similarity function. Alternately, the user may determine the degree to which units of information appear to be similar to each other based on his or her knowledge and preferences.

The disclosed method can be executed using a computer system, such as a personal computer or the like, as is well known in the art. The disclosed system can be a stand-alone system, or it can be incorporated in a computer system, in which case the user interface can be the graphical or other user interface of the computer system and the cluster structure knowledge base can be, for example, a file in any of the computer system's storage areas, elements or devices.

Referring to **FIG. 1**, there is provided a user-configurable information clustering system **10** comprising an *information clustering engine 20*, a *personalization module 50*, a *cluster structure knowledge base 14*, and a *user interface module 16*. Information in the form of information and preference vectors is supplied to the clustering engine **20**, which comprises a hybrid neural network model that performs a hybrid of supervised and unsupervised learning. The neural network may be conventional. For example, an ARTMAP system or an ARAM system, such as described in "Adaptive Resonance Associative Map", published in "Neural Networks", Vol. 8 No. 3, pp. 437-446 (1995), which is incorporated herein by reference, can be used. The user interface module **16** may comprise a graphical user interface, keyboard, keypad, mouse, voice command recognition system, or any combination thereof, and may permit graphical visualization of information groupings. The cluster structure knowledge base **14** may be any conventional recordable storage format, for example a file in a storage device, such as magnetic or optical storage media, or in a storage area of a computer system.

The user-configurable information clustering system **10** allows a user to personalize or influence the cluster structure by indicating his or her own preferences in the form of preference vectors. Through the *user interface module 16* and the *personalization module 50*, the user is able to create a customized cluster structure by selective and/or repeated application of the following: creating a new cluster, labeling an existing cluster, and modifying cluster structure by merging and splitting clusters. In addition, the customized cluster structure can be stored in the *cluster structure knowledge base 14* and retrieved at a later stage for processing new information.

As described in the article cited above, ARAM is a family of neural network models that performs incremental supervised learning of recognition categories (pattern classes) and multidimensional maps of both binary and analog patterns. Referring to **FIG.**

2A, an ARAM system can be visualized as two overlapping Adaptive Resonance Theory (ART) modules consisting of two input fields F_1^a 22 and F_1^b 26 with a cluster field F_2 30.

Each F_2 cluster node j is associated with an adaptive template information vector w_j^a and corresponding adaptive template preference vector w_j^b . Initially, all cluster nodes are uncommitted and all weights are set equal to 1. After a cluster node is selected for encoding, it becomes committed.

Fuzzy ARAM dynamics are determined by the choice parameters $\alpha^a > 0$ and $\alpha^b > 0$; the learning rates β^a in $[0,1]$ and β^b in $[0,1]$; the vigilance parameters ρ^a 24 in $[0,1]$ and ρ^b 28 in $[0,1]$; and a contribution parameter γ in $[0,1]$. The choice parameters α^a and α^b control the bias towards choosing a F_2 cluster whose template information and preference vectors have a larger norm or magnitude. The learning rates β^a and β^b control how fast the template information and preference vectors w_j^a and w_j^b adapt to the input information and preference vectors **A** and **B**, respectively. The vigilance parameters ρ^a and ρ^b determine the criteria for a satisfactory match between the input and the template information and preference vectors, respectively. The contribution parameter γ controls the weighting of contribution from the information and preference vectors when selecting an F_2 cluster.

Referring to **FIG. 2B**, given an information vector **A** with an associated preference vector **B**, the system first searches for an F_2 cluster J encoding a template information vector w_J^a and a template preference vector w_J^b paired therewith that are closest to the input information vector **A** and the input preference vector **B**, respectively, according to a similarity function. Specifically, for each F_2 cluster j , the information clustering engine calculates a similarity score based on the input information and preference vectors **A** and **B**, respectively, and the template information and preference vectors w_j^a and w_j^b , respectively. An example of a similarity function is given below as the category choice function, eqn. (3). The F_2 cluster that has the maximal similarity score is then selected and indexed at J .

Referring to **FIG. 2C**, the information clustering engine performs template matching to verify that the template information vector w_J^a and the template preference vector w_J^b of the selected category J match well with the input information vector **A** and the input preference vector **B**, respectively, according to another similarity function, *e.g.*

eqn. (4) below. If so, the system performs template learning to modify the template vectors w_j^a and w_j^b of the F_2 cluster J to encode the input information and preference vectors A and B , respectively. Otherwise, the cluster is reset and the system repeats the process until a match is found. The detailed algorithm is given below.

5 The ART modules used in ARAM may be of a type which categorizes binary patterns, analog patterns, or a combination of the two patterns (referred to as "fuzzy ART"), as is known in the art. Described below is a fuzzy ARAM model composed of two overlapping fuzzy ART modules.

10 Referring to FIG. 3, the dynamics of the information clustering engine 20 is described as follows. Given a pair of F_1^a and F_1^b input vectors A and B , for example, an information vector and preference vector, respectively, for each F_2 node j , a *category choice process* 32 computes the choice function T_j as defined by

$$T_j = \gamma |A \wedge w_j^a| / (\alpha^a + |w_j^a|) + (1-\gamma) |B \wedge w_j^b| / (\alpha^b + |w_j^b|) \quad (3)$$

15 where, for vectors p and q , the fuzzy AND operation is defined by $(p \wedge q)_i = \min(p_i, q_i)$, and the norm is defined by $|p| = \sum_i p_i$. The system is said to make a choice when at most one F_2 node can become active. The choice is indexed at J by a *select winner process* 34 where $T_J = \max \{T_j: \text{for all } F_2 \text{ nodes } j\}$.

20 A *template matching process* 36 then checks if the selected cluster represents a good match. Specifically, a check 38 is performed to verify if the *match functions*, m_j^a and m_j^b , meet the vigilance criteria in their respective modules:

$$m_j^a = |A \wedge w_j^a| / |A| \geq \rho^a \text{ and } m_j^b = |B \wedge w_j^b| / |B| \geq \rho^b. \quad (4)$$

25 Resonance occurs if both criteria are satisfied. Learning then ensues, as defined below. If any of the vigilance constraints is violated, mismatch reset 42 occurs in which the value of the choice function T_j is set to 0 for the duration of the input presentation. The search process repeats, selecting a new index J until resonance is achieved.

Once the search ends, a *template learning process* 40 updates the template information and preference vectors w_j^a and w_j^b , respectively, according to the equations

$$w_j^{a(\text{new})} = (1-\beta^a) w_j^{a(\text{old})} + \beta^a (A \wedge w_j^{a(\text{old})}) \quad (5)$$

and

$$\mathbf{w}_J^{b \text{ (new)}} = (1 - \beta^b) \mathbf{w}_J^{b \text{ (old)}} + \beta^b (\mathbf{B} \wedge \mathbf{w}_J^{b \text{ (old)}}) \quad (6)$$

respectively. For efficient coding of noisy input sets, it is useful to set $\beta^a = \beta^b = 1$ when J is an uncommitted node, and then take $\beta^a < 1$ and $\beta^b < 1$ after the cluster node is committed. *Fast learning* corresponds to setting $\beta^a = \beta^b = 1$ for committed nodes.

5 At the start of each input presentation, the vigilance parameter ρ_a equals a baseline vigilance ρ^a . If a reset occurs in the category field F_2 , a *match tracking process* 44 increases ρ^a until it is slightly larger than the match function m_J^a . The search process then selects another F_2 node J under the revised vigilance criterion.

Referring to **FIG. 4**, the *personalization module 50* works in conjunction with the 10 *information clustering engine 20* to incorporate user preferences to modify the machine-generated cluster structure.

An exemplary parameter setting for the information clustering engine **20** is as follows: $\alpha^a = \alpha^b = 0.1$, $\beta^a = \beta^b = 1$, $\rho^a = 0.5$, $\rho^b = 1$, and $\gamma = 0.5$. During automatic clustering, no user preference is given, and the information clustering engine **20** automatically generates 15 a cluster structure, referred to as a machine-generated cluster structure: For each unit of information (Φ), a pair of vectors (\mathbf{A} , \mathbf{B}_0) is presented to the system, where \mathbf{A} is the representation vector of Φ and

$$B_{0i} = 1 \text{ for } i = 1, \dots, 2N. \quad (7)$$

Since $|\mathbf{B}_0 \wedge \mathbf{w}_J^b| / |\mathbf{B}_0|$ equals 1, condition (6) reduces to

$$20 \quad m_J^a = |\mathbf{A} \wedge \mathbf{w}_J^a| / |\mathbf{A}| \geq \rho^a. \quad (8)$$

Essentially, the system now operates like a pure clustering system that self-organizes the information based on similarities in the information vectors. The coarseness of the information groupings is controlled by the baseline vigilance parameter (ρ^a).

A *create cluster module 52* allows the user to add a new information cluster into 25 the system so that information can be organized according to such an information grouping. Through the *user interface module 16*, the user can input a pair of template information and preference vectors (\mathbf{w}_J^a , \mathbf{w}_J^b) which defines the key attributes of the information in the cluster together with a cluster label, if any. The resulting clusters

reflect the user's preferred way of grouping information and can be used as the default slots for organizing information.

A *label cluster module 54* allows the user to assign labels to "mark" certain information groupings that are of particular interest (to the user) so that new information can be organized according to such information groupings. Through the *user interface module 16*, the user can assign a label L_k to a cluster j by modifying the template preference vector w_j^b to equal B_k , where B_k is a preference vector representing L_k . Labels reflect the user's interpretation of the groupings. They are useful landmarks to the user in navigating the information database and locating old as well as new information.

Using the *label cluster module 54*, the user is able to merge clusters implicitly by labeling them with the same labels. In this case, the merging is said to be a local one as it only affects the clusters that are labeled. To do a global merging, a *merge cluster module 56* allows the user to combine two or more information groupings generated by the clustering process using a lower vigilance parameter value. Through the *user interface module 16*, the user can select one or more units of information in each of two different clusters as an indicative standard of similarity. As an example, A_1 and A_2 can be two information vectors representing two units of information. The revised baseline vigilance parameter ρ^a would then be computed as

$$\rho^a = \min (|A_1 \wedge A_2| / |A_1|, |A_1 \wedge A_2| / |A_2|). \quad (9)$$

Using the new baseline vigilance parameter, A_1 and A_2 will satisfy the match condition as stated in (4) and may be grouped into one cluster as a result of the relaxed similarity criteria. The effect of cluster merging is global, in the sense that the system now operates at a lower vigilance on the whole, grouping items together that it would otherwise distinguish.

A *split cluster module 58* allows the user to split an information group into two or more clusters by indicating that certain units of information are sufficiently different to be grouped separately. Through the *user interface module 16*, the user can select two specific units of information, for example A_1 and A_2 , in a cluster and assign to them two different labels, for example L_1 and L_2 , represented by B_1 and B_2 , respectively.

The updated pairs of information and preference vectors, (A_1, B_1) and (A_2, B_2) , together with the rest of the vectors, are presented to the information clustering engine **20** for re-clustering. Since $B_1 \neq B_2$, and $\rho^b=1$, A_1 and A_2 will be grouped into different clusters. In addition, the remaining information vectors, originally in a single cluster, will
 5 be re-organized into one of the two new clusters based on their similarities to A_1 and A_2 .

In another example, A_1 and A_2 can be used to tighten the match condition for the entire information space. In this case, the baseline vigilance parameter ρ^a would be re-computed as

$$\rho^a = \max (|A_1 \wedge A_2| / |A_1|, |A_1 \wedge A_2| / |A_2|) + \epsilon, \quad (12)$$

10 where ϵ is a small constant.

By selective and/or repeated application of the create cluster, label cluster, merge cluster and split cluster functions of the personalization module **50**, the user is able to create a labeled cluster structure which incorporates his own preferences.

To preserve the labeled cluster structure for a fresh clustering session, a *store cluster module* **60** transfers the following information to the cluster structure knowledge
 15 base **14**.

1. The baseline vigilance parameter (ρ^a).
2. For each labeled cluster j , the template information vector w_j^a and the associated
 20 template preference vector w_j^b .

The stored clusters and cluster structure can be retrieved at a later stage from the knowledge base **14** using a *retrieve cluster module* **62** to initialize the architecture of the information clustering engine **20**, that is, the retrieved cluster structure is used as the
 25 initial structure of the information clustering engine. Based on the initialized cluster structure, new information can be organized according to the user's preferences stored over the previous sessions. The cluster structure may also be modified by further personalization.

An embodiment of the disclosed invention is *personalized document navigation* wherein the user is allowed to customize the document navigation space, *i.e.* the organized collection of information available to the user, with respect to his or her interpretation and preferences.

5 Another embodiment of the invention is a *drag-and-draw approach to building a categorization system*. Information is first automatically clustered into natural groupings based on similarities in content. By modifying the machine-generated groupings, the user can define her preferred groupings using the personalization functions as described herein. A classification system can be created in an intuitive and interactive manner
10 without the need for example annotation (*i.e.*, labeling information) or knowledge specification (*e.g.*, written classification rules).

Yet another embodiment is *detection of new information and trend analysis*. The user defines his know-how and interpretation of the environment in terms of how he wants the information to be organized. New information is supplied periodically to the
15 information clustering system. New information that falls within the user-defined cluster structure corresponds to familiar themes of information. Any new information that falls outside of the defined cluster structure represents new themes which are potentially interesting to the user. Repeated personalization of the information in the information database, *i.e.* creating, labeling, merging, splitting, and storing clusters and the resulting
20 labeled cluster structure, helps the user to identify information that is new with respect to his experience and to analyze unexpected trends.

Various preferred embodiments of the invention have now been described. While these embodiments have been set forth by way of example, various other embodiments and modifications will be apparent to those skilled in the art. Accordingly, it should be
25 understood that the invention is not limited to such embodiments, but encompasses all that which is described in the following claims.